

# Rajalakshmi Engineering College

Name: Manishraj A  
Email: 240801192@rajalakshmi.edu.in  
Roll no: 240801192  
Phone: 9514051184  
Branch: REC  
Department: I ECE AF  
Batch: 2028  
Degree: B.E - ECE

Scan to verify results



## NeoColab\_REC\_CS23231\_DATA STRUCTURES

### REC\_DS using C\_Week 1\_COD\_Question 5

Attempt : 1  
Total Mark : 10  
Marks Obtained : 10

#### Section 1 : Coding

##### 1. Problem Statement

Imagine you are tasked with developing a simple GPA management system using a singly linked list. The system allows users to input student GPA values, insertion should happen at the front of the linked list, delete record by position, and display the updated list of student GPAs.

##### ***Input Format***

The first line of input contains an integer  $n$ , representing the number of students.

The next  $n$  lines contain a single floating-point value representing the GPA of each student.

The last line contains an integer position, indicating the position at which a student record should be deleted. Position starts from 1.

### **Output Format**

After deleting the data in the given position, display the output in the format "GPA: " followed by the GPA value, rounded off to one decimal place.

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 4

3.8

3.2

3.5

4.1

2

Output: GPA: 4.1

GPA: 3.2

GPA: 3.8

### **Answer**

```
// You are using GCC
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
struct node{  
    float data;  
    struct node *next;  
}*head=NULL;
```

```
typedef struct node node;
```

```
void insert(float value){  
    node *newnode=(node*)malloc(sizeof(node));  
    newnode->data=value;  
    newnode->next=head;  
    head=newnode;  
}
```

```
void display(){
```

```

    node *temp=head;
    while(temp!=NULL){
        printf("GPA: %.1f",temp->data);
        printf("\n");
        temp=temp->next;
    }
    printf("\n");
}

```

```

int lenght()

```

```

{
    int count =0;
    node *temp=head;
    while(temp!=NULL){
        count++;
        temp=temp->next;
    }
    return count;
}

```

```

void delbeg(){

```

```

    if(head!=NULL){
        node *tempnode=head;
        head=head->next;
        free(tempnode);
    }
}

```

```

void delend(){

```

```

    if(head==NULL)
        return;
    if(head->next==NULL){
        free(head);
        return;
    }
    node* temp=head;
    while(temp->next->next!=NULL)
        temp=temp->next;
    temp->next=NULL;
}

```

```

void delmid(int position){

```

```

    if(position==1){
        delbeg();
        return;
    }
    node* temp=head;
    node* p=NULL;
    int count=1;
    while(temp!=NULL&&count<position){
        p=temp;
        temp=temp->next;
        count++;
    }
    if(temp==NULL)
        return;
    p->next=temp->next;
    free(temp);
}

```

```

void del(int pos){
    int n=lenght();
    if(pos==1){
        delbeg();
        display();
    }
    else if(pos==n){
        delend();
        display();
    }
    else {
        delmid(pos);
        display();
    }
}

```

```

int main(){
    int n,a;
    float values;
    scanf("%d",&n);
    for(int i=0;i<n;i++){
        scanf("%f",&values);
        insert(values);
    }
}

```

```
}  
scanf("%d",&a);  
del(a);  
}
```

**Status :** Correct

**Marks :** 10/10